Content - Overview

Motivation - Introduction
- Diagnostic-Therapeutic Cycle
- Medical Therapy Planning

Guideline Repositories

Guideline Development (CBO)

Instruments for Quality of Guidelines
- Agree Instrument
- (e)GLIA

Planning / Plan Management

Part 1

Planning ↔ Plan Management

Overview

Planning
- Definitions
- Types
- STRIPS-based planning

From Planning to Plan Management
- Definition
- Tasks
- Applications and Examples

Definition: Planning

[Russell & Norvig, 1995]

Given:

I : Initial state
(current world state)

A: a set of action definitions

G: Goal
(desired world state)

Determine:
- Action selection
- Parameter selection
- Action ordering
- Resource allocation

a sequence of actions (a_1 .. a_n) when executed beginning in *I* results in a state where *G* is true

Definition: Scheduling

[Russell & Norvig, 1995]

Given:

a set of activities
$(a_1 .. a_n)$

a set of activity requirements
$(a_{1,1} .. a_{n,j})$

- resource requirements
- state requirements
- temporal relations among activities

Determine:

for each activity a_i

- a start and end time
  $(a_{i,\,start}\ a_{i,end})$
  that
  resource, state temporal requirements of each a_i are met

Planning vs. Scheduling

[Russell & Norvig, 1995]

Planning focuses more on

- action selection
- action ordering

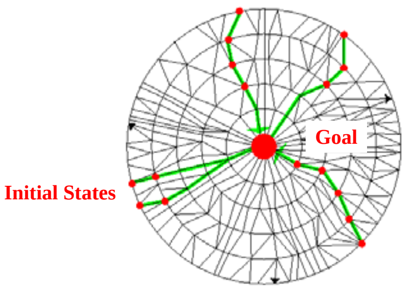
Scheduling focuses more on

- resource assignments
- exact timing
- methods used in Operation Research (OR)
  e.g., PERT charts, critical paths

Planner: Search Types

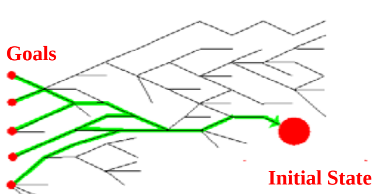
Progression planner

- from initial state to goal
  - e.g., POP-Planner



Regression planner

- from goal to initial state
  - e.g., Graphplan



Strategies of Planner

Linear Planning

Non-Linear Planning

Regression Planning

Progression Planning

Hierarchical Planning

Reactive Planning

Mixed-Initiative Planning

Example: Shopping Problem

Initial state

At(Home, S_0) ∧ ¬ *Have(Milk, S_0)* ∧ ¬ *Have(Bananas, S_0)*
∧ ¬ *Have(Drill, S_0)*

Operators (set of action)

$\forall a,s$ *Have(Milk, Result(a,s))* ⇔ [*(a=Buy(Milk)* ∧ *At(Supermarket, s)*
∨ *(Have(Milk, s)* ∧ *a* ≠ Drop(Milk)]

Goal state

∃ s *At(Home, s)* ∧ *Have(Milk, s)* ∧ *Have(Bananas, s)* ∧
Have(Drill, s)

STRIPS-based Planning

[Fikes, Nilsson 1971]

STanford Research Institute Problem Solver

Representations for Actions

Action description

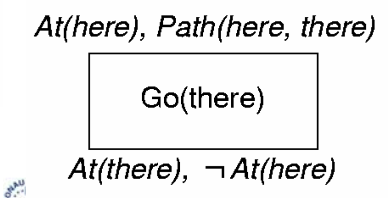
Precondition

conjunction of atoms (positive literals), must be true before the operator can be applied

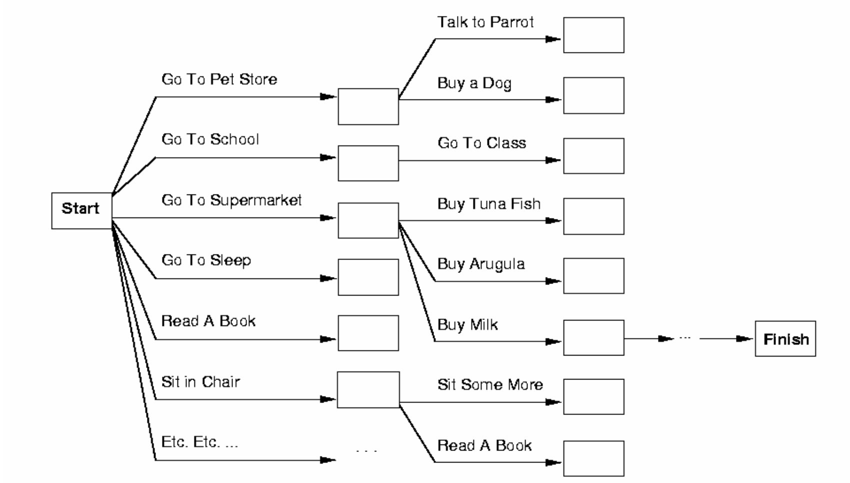
Effects

ADD-lists and DELETE-lists -> no Frame-Problem!

$Op($Action: $Go(there)$,
Precond: $At(here) \land Path(here, there)$,
Effect: $At(there) \land \neg At(here))$

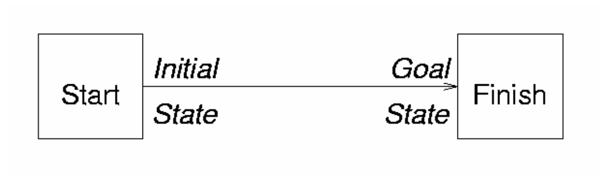


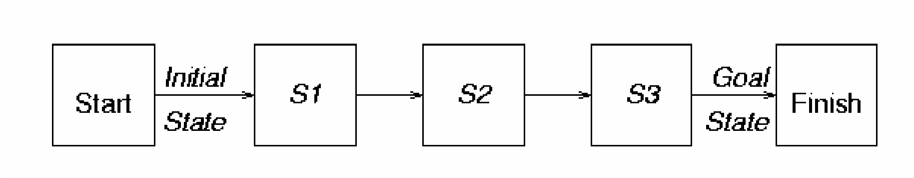
Search Space (State)



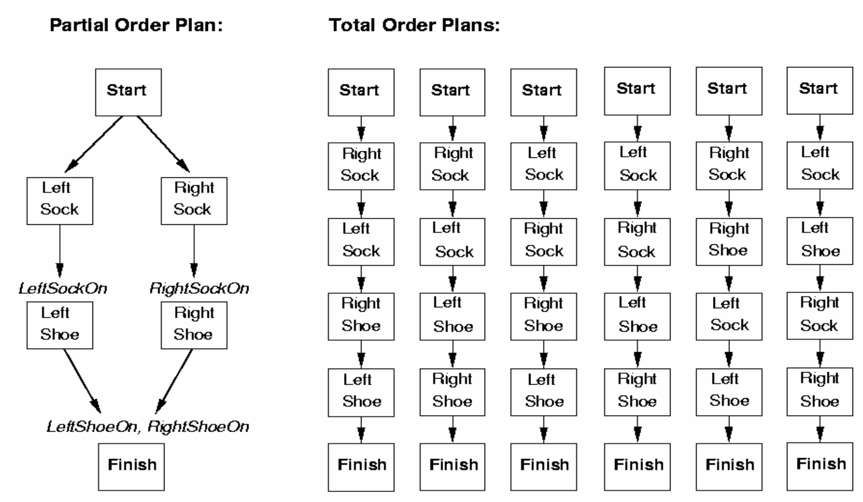
Search Space (Plan)

Partial plan





Plan = Sequence of Actions



Plan Representation

A set of plan steps

STRIPS operators

A set of step ordering constraints

$S_i < S_j$

A set of variable binding constraints

x = t, x .. variable, t= constant or variable

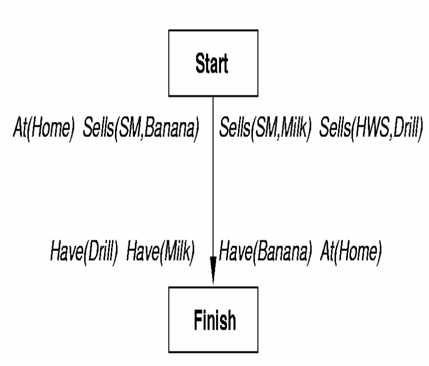
A set of causal links (protection intervals)

"S_i achieves c for S_j"

$$S_i \xrightarrow{c} S_j$$

Solution: complete and consistent plan

Example: Shopping Problem



Op (Action: *go(there)*,
Precond: *At(here)*,
Effect: *At(there)* ∧ ¬*At(here)*)

Op (Action: *buy(x)*,
Precond: *At(store)* ∧ *Sells(store,x)*,
Effect: *Have(x)*)

there, here, x are variables

Classical Planning Extension

[Tsamardinos, Pollack 2000]

Actions with stochastic outcomes
[Kushmerick et al., Goldman/Boddy, Blythe].

Imperfect knowledge
[Etzioni, Peot/Smith, Collins/Pryor].

Non-categorical goals (rich utility functions) [Boutilier, Dean/Kaelbling/Littman, Haddawy, Williamson, Goodwin, Onder/Pollack].

Rich temporal and resource constraints [Bacchus/Kabanza, Ghallab, Muscettola, Pollack/Tsamardinos].

Hierarchical decomposition
[Erol, Tsuneto, Young/Pollack/Moore].

**But all, still form a plan
from a complete set of goals.**

From Planning to Plan Management

Definition
Tasks
Applications and Examples

Research On Planning

[Pollak, Horty 1999]

6 Simplifying Assumptions

Planning agent is omniscient

Actions = definite outcomes

Categorical Goals

Agent is the only source of change in the environment

Goals remain unchanged

Actions can be modeled as instantaneous state transducers

Plan Representation

requirements

Huge volume of data

Pre- and **post-conditions** are needed to control the execution of durative actions or plans

Goal may not be **achievable in time**

Sequential, **parallel**, and **cyclical** execution of plans is necessary

Plan Representation

missing requirements

Large domain knowledge is available

Incomplete and non-deterministic information

Planning agent is **not omniscient**

Unobservable underlying processes

Goals are not categorical and unchanged over time

Everything is **durative**, not instantaneous

Multiple time lines based on different granularities

Plans can be **suspended**

Interleaving of plan design and execution need

Representations of Clinical Protocols

Free Text

+ Easy to write

– Partly vague & incomplete:
  - Intentions
  - Temporal representation
  - etc.

– Transforming into a formal and structured framework

Flow diagrams

+ Medical experts are used to work with
+ Sequential states & actions in a graphical way

– All possible orders of plan execution
– All the exception conditions
– Cover a small subset of the possible situations and possible paths through
– Kinds of layering, which avoid to cope with
  - concurrent (parallel) actions
  - different temporal dimensions
  - high numbers of possible transitions
  - mutual dependencies of parameters

Why Time-Oriented Skeletal Plans?

Definition: Skeletal Plans are plan schemata at various levels of detail that capture the essence of the procedure, but leave room for execution-time flexibility in the achievement of particular goals [Friedland & Iwasaki, 1985].

Representation & reuse of domain-specific procedural knowledge

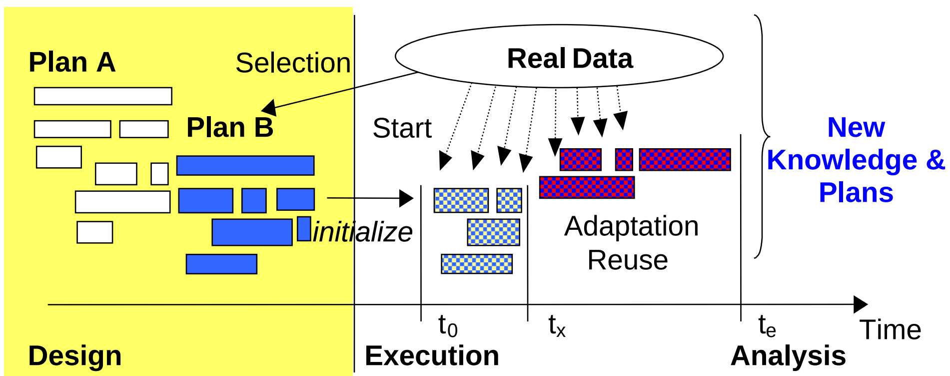
Reusable in different contexts

Automated reactive planners: e.g.,

ONCOCIN [Tu, et al., 1989]
SPIN [Uckun, 1994]

Time-Oriented, Skeletal Plans

Extensions of [Friedland & Iwasaki, 1985]

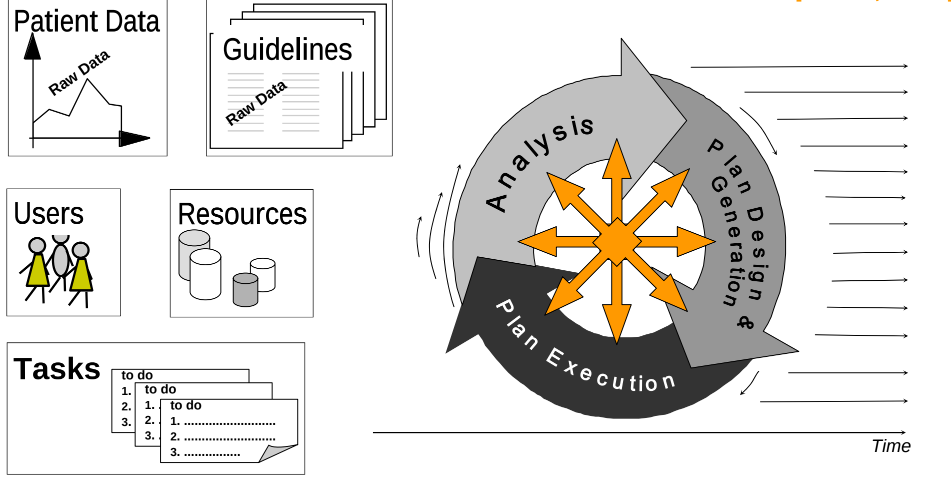


Representation & reuse of domain-specific procedural knowledge

Reusable in different contexts

Which Plan Management is Needed?

[Miksch, 1999]



Fully Intertwined and Interactive Tasks

Plan Management

[Miksch, 1999]

at Design Time

Plan Generation
Advanced Plan Editing
Plan Verification
Plan Validation
Plan Visualization
Plan-Scenario Testing

at Execution Time

Plan Selection
Plan Instantiation
Data Abstraction
Monitoring
Plan Execution
Execution Visualization
Critiquing / Evaluation
Plan Rationale / History

Plan Management: Tasks Mostly Done at **Design** Time

[Miksch, 1999]

Plan Generation	starts from an initial state description and creates a path of activities to reach the desired goal (progressive way), or starting from the goal (regressive way)
Advanced Plan Editing	provides guided support to author plans and helps to browse plans or a plan hierarchy
Domain-Specific Annotations	provides structured support to write domain assumptions or domain activities
Plan Verification	examines the correctness of interrelated clinical plans by a three-level detection of anomalies (*method semantics*)
Plan Validation	compares the intended states against the prescribed actions and intended plans (*domain semantics*)
Plan-Scenario Testing	applies scenarios of plans to test their functionalities and their course of activities
Plan Visualization	communicates efficiently sets of plans to domain experts

Plan Management: Tasks Mostly Done at **Execution** Time

[Miksch, 1999]

(1/2)

Plan Selection	chooses applicable clinical plans from the plan library according to the patient's state, the plan's overall intentions, and plan effects
Plan Instantiation	adjusts the parameters of a clinical plan according to the patient data record and the medical environment
Data Abstraction	transforms information obtained from sensors or user input into a format suitable for the monitoring module, which consists of three tasks: (a) data validation, (b) calculation of derived values, and (c) transformation of time-stamped data into qualitative information.
Plan Monitoring	compares assumptions with reality

Plan Management: Tasks Mostly Done at **Execution** Time

[Miksch, 1999]

(2/2)

Plan Execution	maps plans and actual situations in the medical environment which is done on three distinct layers: (a) plan synchronization, (b) plan adaptation, and (c) replanning.
Plan Modification / Alternatives	handles changes in the environment
Plan Evaluation / Critiquing	analyzes executed plans or plan hierarchies according to their goals and intentions
Plan Visualization	supervises and communicates plans or plan hierarchies
Plan Rationale / History	explains executed plans or plan hierarchies

Tasks in Continual Planning

[Miksch, et a. 2000/2001]

Environment Monitoring

Actual States of the World

No Perfect Model of Reality

Plan Adaptation

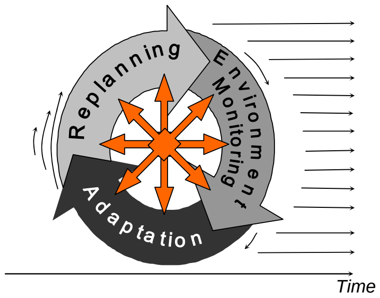
Classes of Exceptions

Prepared Alternatives

Replanning

Unexpected Changes

Merging with History



Plan Management Capabilities

[Pollak, Horty 1999]

Plan Generation

What actions achieve your goal?

Plan Elaboration

How much detail should you include, and when should you add detail?

Commitment

When should you be willing to reconsider your existing plans?

Environment Monitoring

What new opportunities and problems should you attend to?

Alternative Assessment

What's the value of an alternative in context?

Coordination/Cooperation

How should you interact with other agents?

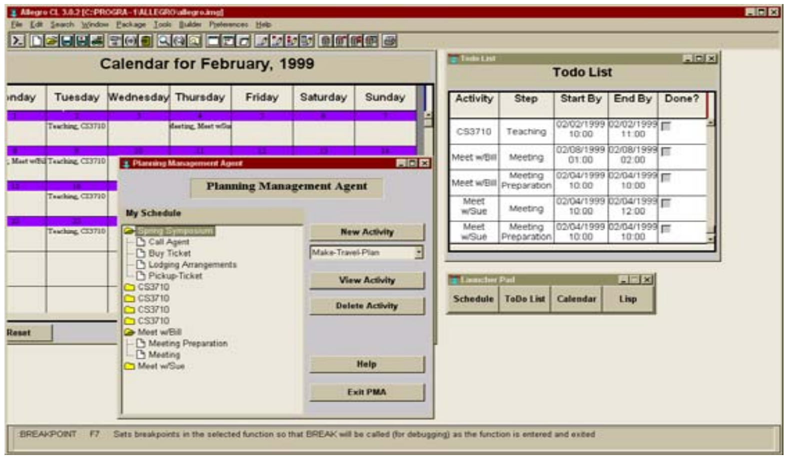
Meta-level Control:

How much effort should you put into planning and evaluation tasks?

Applications

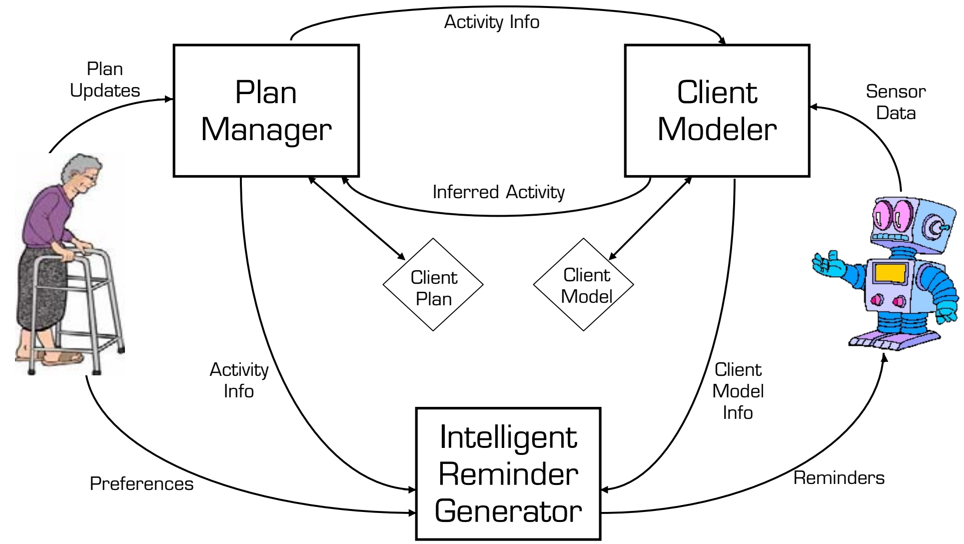
PMA: Plan Management Agent

[Tsamardinos, Pollack 2000]



Scheduling Management: Autominder

[Pollack, 2005]



Plan Manager: What should the client do?

[Pollack, 2005]

Maintains Up-to-Date Record of Client's Planned Activities

Eating, Hydrating, Toileting, Medicine-taking, Exercise, Social Activities, Doctor's Appointments, etc.

Updates Plan & Propagates Constraint

- New Planned Activity Added
- Existing Activity Modified or Deleted
- Planned Activity Performed
- Critical Time Bounds Passed

Models Plans as Disjunctive Temporal Problems & uses AI Planning and CSP Technology for Updating

Client Modeler: What is the client doing?

[Pollack, 2005]

Given Information

- Sensor Input: Client Moved to Kitchen
- Clock Time: at 7:23 a.m.
- Client Plan: Breakfast should be Eaten Between 7 and 8
- Model of Previous Actions: Client has not Yet Eaten Breakfast
- Learned Patterns: 82% of the Time, Client Starts Breakfast between 7:10 and 7:25
- Reminder Information: We Issued a Reminder at 7:21

Infers Probability that Various Events have Occurred

e.g., Client Has Begun Breakfast

Bayesian Reasoning Technology

Addressing Limitations of Previous Approaches to Handle Complex and Dynamic Temporal Relations

Intelligent Reminder Generation

[Pollack, 2005]

Given a Client's Plan and its Execution Status

- Easy to Generate Reminders
  - Remind at Earliest Possible Time of Each Action
- Harder to "Remind Well"
  - Maximize Likelihood of Appropriate Performance of Key Activities
  - Facilitate Efficient Performance
  - Avoid Annoying Client
  - Avoid Making Client Overly Reliant

Local Search Tools

- Incrementally Refine Reminder Plans
- Investigating Reinforcement Learning For Adaptive Interaction Policies

PEARL …

[Thrun, 2005]

PEARL

[Thrun, 2005]

PEARL

[Thrun, 2005]

Comparison With Related Paradigms

[Tsamardinos, Pollack 2000]

Reactive Systems: (PRS, RAPS, etc.)

Do not project current commitments to the future and therefore cannot identify conflicts.

Limited causal information.

Rich plan language but limited temporal constraints.

No cost in context capabilities.

Calendar Systems: (Microsoft Outlook)

Have explicit time, but can only schedule simple events.

Interactions limited to busy and free time slots, extremely limited temporal constraints.

Plan Management

Rich plan language including temporal constraints and contingencies (conditions).

Evaluates cost in context.

Projects current plans into the future, identifies conflicts, and suggests solutions.

Classical Planning Systems:

Solve one problem at a time, no agenda with current commitments.

Limited expressiveness of plans.

Workflow Systems:

Limited capabilities for handling temporal uncertainty and contingencies.

Limited plan interaction and threat resolution capabilities.

No reasoning about value of alternative ways to perform a task.